IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): DePauw et al.
Docket No.: YOR920010309US2
Serial No.: 10/040,344
Filing Date: January 2, 2002
Group: 2193
Examiner: Jason D. Mitchell

Title: Method and Apparatus for Tracing Details of a Program Task

---

TRANSMITTAL OF REPLY BRIEF

Mail Stop Appeal Brief - Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith are the following documents relating to the above-identified patent application:

(1) Reply Brief.

In the event of non-payment or improper payment of a required fee, the Commissioner is authorized to charge or to credit **IBM Corporation Deposit Account No. 50-0510** as required to correct the error. A duplicate copy of this letter is enclosed.

Respectfully,

Kevin M. Mason
Attorney for Applicant(s)
Reg. No. 36,597
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06824
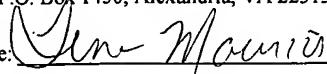(203) 255-6560

Date: July 17, 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

5   Applicant(s):   DePauw et al.
      Docket No.:   YOR920010309US2
      Serial No.:   10/040,344
      Filing Date: January 2, 2002
      Group: 2193
10   Examiner: Jason D. Mitchell

      Title: Method and Apparatus for Tracing Details of a Program Task

15   **REPLY BRIEF**

Mail Stop Appeal Brief – Patents
Commissioner for Patents
20   P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
25

      Appellants hereby reply to the Examiner's Answer, mailed May 16, 2006 (referred to hereinafter as "the Examiner's Answer"), in an Appeal of the final rejection of claims 1-35 in the above-identified patent application.

30   **REAL PARTY IN INTEREST**

      A statement identifying the real party in interest is contained in Appellants' Appeal Brief.

                     RELATED APPEALS AND INTERFERENCES

35   A statement identifying related appeals is contained in Appellants' Appeal Brief.

STATUS OF CLAIMS

A statement identifying the status of the claims is contained in Appellants' Appeal Brief.

5

STATUS OF AMENDMENTS

A statement identifying the status of the amendments is contained in Appellants' Appeal Brief.

SUMMARY OF CLAIMED SUBJECT MATTER

10   A Summary of the Invention is contained in Appellants' Appeal Brief.

STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A statement identifying the grounds of rejection to be reviewed on appeal is contained in Appellants' Appeal Brief.

15

CLAIMS APPEALED

A copy of the appealed claims is contained in an Appendix of Appellants' Appeal Brief.

20   ## ARGUMENT

In the Response to Arguments section of the Examiner's Answer, the Examiner asserts that Laffra's "hooks 'pass information...each time a method...is entered or left' clearly disclosing 'conditions to initiate a trace' as claimed."

The present disclosure teaches that, "according to another aspect of the

25   invention, discussed further below in conjunction with FIGS. 3 and 4, for each program task under analysis, *the user can define what commences a task and what concludes a task.* Thus, the software analysis tool 100 will *begin tracing only when the user-specified conditions for commencing a task are present.*" (Page 5, lines 1-5; emphasis added.)

As Appellants have noted regarding the visualization script cited by the

30   Examiner, Laffra teaches that

the script 285 will tell the monitoring function (FIG. 3) *how to **interpret** the information which is generated by the hooks 260 and 270.* The visualization script interpretation process is described in FIG. 4.
(Col. 5, lines 39-42; emphasis added.)

Laffra also teaches that

the monitoring function receives *information generated by hooks 260 and 270, when they are executed at runtime. The information that is gathered by the monitoring function is then visualized on a graphics display, guided by the set of rules 288, to be found in the visualization script 285.* Each time a particular hook is executed, the monitoring function 300 will inspect the current display and the script. The monitoring function 300 then modifies the display depending on the hook and the visualization script.
(Col. 5, line 65, to col. 6, line 7; emphasis added.)

Laffra does ***not*** disclose or suggest that the hooks 260 and 270 are *conditional instructions*, and thus a person of ordinary skill in the art would recognize that hooks 260 and 270 are executed whenever they are encountered. Laffra also does ***not*** disclose or suggest that the script 285 determines or controls ***when** information is generated by the hooks 260 and 270*, and does ***not*** disclose or suggest that the script 185 determines or controls *what information is generated by the hooks 260 and 270*. Laffra therefore does not disclose or suggest utilizing *one or more **conditions** to **initiate** a trace* of a program task. Independent claims 1, 24, and 32-35 require collecting details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one *or more **conditions** to **initiate** a trace* of said program task or monitoring said software system to identify said program task based on a specification associated with said program task, wherein said specification contains *one or more **conditions** to **initiate** a trace* of said program task.

Thus, Laffra does not disclose or suggest collecting details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task, as required by independent claims 1, 32, and 34, and does not disclose or suggest monitoring said software system to identify said program

3

task based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task, as required by independent claims 24, 33, and 35.

<u>Claims 2-4, 6-8 and 25</u>

5   In the Response to Arguments section of the Examiner's Answer, the Examiner asserts that Laffra's hooks clearly define a duration of a program task by indicating its start and end points (entered or left).

As noted above, the present disclosure teaches that, "according to another aspect of the invention, discussed further below in conjunction with FIGS. 3 and 4, for

10   each program task under analysis, *the user can define what commences a task and what concludes a task.* Thus, the software analysis tool 100 will *begin tracing only when the user-specified conditions for commencing a task are present.*" (Page 5, lines 1-5; emphasis added.)

Appellants have reviewed the citations presented by the Examiner, but

15   could find no disclosure or suggestion by Laffra that a ***duration of a program task*** is defined by said *one or more **conditions associated with a state of the software system***, that said *one or more **conditions** includes an entry or exit of at least one specified method*, or that said *one or more **conditions** includes a creation or deletion of at least one specified object*. Claims 2 and 25 require wherein a duration of said program task is

20   defined by said one or more conditions associated with a state of said software system. Claim 3 requires wherein said one or more conditions includes an entry or exit of at least one specified method. Claim 4 requires wherein said one or more conditions includes a creation or deletion of at least one specified object.

Appellants also could find no disclosure or suggestion by Laffra that

25   "assigning new values to global variables or local variables" is a ***condition*** *that defines a duration of said program task,* or *that **conditions** include at least one specified resource exceeding at least one specified threshold.* Claim 6 requires wherein said one or more conditions includes a passing of at least one specified object or scalar value as an argument, return value or field value that defines a duration of said program task. Claim

30   8 requires wherein said one or more conditions includes at least one specified resource

exceeding at least one specified threshold.

Thus, Laffra does not disclose or suggest wherein a duration of said program task is defined by said one or more conditions associated with a state of said software system, as required by claims 2 and 25, does not disclose or suggest wherein

5   said one or more conditions includes an entry or exit of at least one specified method, as required by claim 3, does not disclose or suggest wherein said one or more conditions includes a creation or deletion of at least one specified object, as required by claim 4, does not disclose or suggest wherein said one or more conditions includes a passing of at least one specified object or scalar value as an argument, return value or field value that

10   defines a duration of said program task, as required by claim 6, and does not disclose or suggest wherein said one or more conditions includes at least one specified resource exceeding at least one specified threshold, as required by claim 8.

Conclusion

15   The rejections of the cited claims under section 102 in view of Laffra et al. are therefore believed to be improper and should be withdrawn. The remaining rejected dependent claims are believed allowable for at least the reasons identified above with respect to the independent claims.

The attention of the Examiner and the Appeal Board to this matter is

20   appreciated.

Respectfully,

Kevin M. Mason

25   Date: July 17, 2006   Kevin M. Mason
                                                  Attorney for Applicant(s)
                                                  Reg. No. 36,597
                                                  Ryan, Mason & Lewis, LLP
                                                  1300 Post Road, Suite 205
30   Fairfield, CT 06824
                                                  (203) 255-6560

APPENDIX

1.   A method for analyzing behavior of a software system, comprising: collecting details associated with a program task associated with said software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and providing said collected details for analysis.

2.   The method of claim 1, wherein a duration of said program task is defined by said one or more conditions associated with a state of said software system.

3.   The method of claim 2, wherein said one or more conditions includes an entry or exit of at least one specified method.

4.   The method of claim 2, wherein said one or more conditions includes a creation or deletion of at least one specified object.

5.   The method of claim 2, wherein said one or more conditions includes an invocation of at least one specified object.

6.   The method of claim 2, wherein said one or more conditions includes a passing of at least one specified object or scalar value as an argument, return value or field value.

7.   The method of claim 2, wherein said one or more conditions includes at least one specified sequence of method invocations.

8.   The method of claim 2, wherein said one or more conditions includes at least one specified resource exceeding at least one specified threshold.

9.   The method of claim 1, wherein said collected details include an existence or sequence of specified method invocations.

10.   The method of claim 1, wherein said collected details include an existence or sequence of specified object creations and deletions.

11.   The method of claim 1, wherein said collected details include an existence or sequence of specified class loading and unloading.

12.   The method of claim 1, wherein said collected details include values of specified arguments to invocations of specified methods.

13.   The method of claim 1, wherein said collected details include values of specified return values from invocations of specified methods.

14.   The method of claim 1, wherein said collected details include values of specified field values for invoked objects or field values for passed arguments.

15.   The method of claim 1, further comprising the step of collecting said details for at least one specified number of task instances.

16.   The method of claim 1, further comprising the step of collecting said details for at least one specified number of threads.

17.   The method of claim 1, further comprising the step of dynamically modifying said specification associated with said program task associated with said analysis in an iterative process.

18.   The method of claim 1, further comprising the step of dynamically modifying said specification to identify which details to collect in an iterative process.

19.   The method of claim 1, further comprising the step of connecting to a running version of said software system.

20.   The method of claim 1, further comprising the step of visually analyzing said collected details.

21.   The method of claim 1, further comprising the step of visually analyzing said collected details for a plurality of instances of said program task.

22.   The method of claim 1, further comprising the step of quantitatively analyzing said collected details.

23.   The method of claim 1, further comprising the step of quantitatively analyzing said collected details for a plurality of instances of said program task.

24.   A method for tracing details associated with a program task executing in a software system, comprising:

monitoring said software system to identify said program task based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and

providing trace details associated with said program task.

25.   The method of claim 24, wherein a duration of said program task is defined by said one or more conditions associated with a state of said software system.

26.   The method of claim 25, wherein said one or more conditions is selected from the group consisting essentially of (i) an entry or exit of at least one specified method, (ii) a creation or deletion of at least one specified object, (iii) an invocation of at least one specified object, (iv) a passing of at least one specified object or scalar value as an argument, return value or field value, (v) at least one specified sequence of method

invocations, and (vi) at least one specified resource exceeding at least one specified threshold.

27.   The method of claim 24, wherein said collected details include at least one of the following: (i) an existence or sequence of specified method invocations, (ii) an existence or sequence of specified object creations and deletions, (iii) an existence or sequence of specified class loading and unloading, (iv) values of specified arguments to invocations of specified methods; (v) values of specified return values from invocations of specified methods, and (v) values of specified field values for invoked objects or field values for passed arguments.

28.   The method of claim 24, further comprising the step of collecting said details for at least one of at least one specified number of task instances and at least one specified number of threads.

29.   The method of claim 24, further comprising the step of dynamically modifying a specification associated with said program task associated with said analysis in an iterative process.

30.   The method of claim 24, further comprising the step of dynamically modifying said specification to identify which details to collect in an iterative process.

31.   The method of claim 24, further comprising the step of connecting to a running version of said software system.

32.   A system for analyzing behavior of a software system, comprising:
          a memory that stores computer-readable code; and
          a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:
               collect details associated with a program task associated with said

software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and

provide said collected details for analysis.

5   33.   A system for tracing details associated with a program task executing in a software system, comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

10   monitor said software system to identify said program task based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task;and

provide trace details associated with said program task.

15   34.   An article of manufacture for analyzing behavior of a software system, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to collect details associated with a program task associated with said

20   software system based on a specification associated with said program task, wherein said specification contains one or more conditions to initiate a trace of said program task; and

a step to provide said collected details for analysis.

35.   An article of manufacture for tracing details associated with a program

25   task executing in a software system, comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to monitor said software system to identify said program task based on a specification associated with said program task, wherein said specification contains

30   one or more conditions to initiate a trace of said program task; and

10

a step to provide trace details associated with said program task.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to § 1.130, 1.131, or 1.132 or entered by the Examiner and relied upon by appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 CFR 41.37.